Association for Information Systems
AIS Electronic Library (AISeL)

All Sprouts Content | Sprouts

4-10-2008

Using Interface Design Standards to Support Pervasive Computing

Matt Germonprez
University of Wisconsin - Eau Claire, germonr@uwec.edu

Nikhil Srinivasan
Case Western Reserve University

Michel Avital
Case Western Reserve University, michel@avital.net

芽|Sprouts Working Papers on Information Systems ISSN 1535-6078

Using Interface Design Standards to Support Pervasive Computing

Matt Germonprez
University of Wisconsin - Eau Claire, USA

Nikhil Srinivasan
Case Western Reserve University, USA

Michel Avital
Case Western Reserve University, USA

Abstract

By definition, pervasive computing should support endless expressions of varying interfaces, contexts and automation. As pervasive computing becomes more prevalent, it is important that designers build systems in support of unique, in-use and user-defined characteristics. We explore this expressiveness and propose the use of existing technologies to enhance it. We investigate the impact of interface design standards on two essential aspects of pervasive computing: device independence and usability. Our findings suggest that the application of proper interface design standards can improve data delivery across independent devices with varied bandwidth and resource availability, thereby providing device independence and improved usability respectively. We demonstrate that through their effect on device independence and usability, interface design standards play an important role in the evolution, expansion and expressiveness of pervasive computing.

Keywords: Standards, Pervasive Computing, User Interface, Hypertext

Permanent URL: http://sprouts.aisnet.org/5-14

Reference: Germonprez, M., Srinivasan, N., Avital, M. (2005). "Using Interface Design Standards to Support Pervasive Computing," Case Western Reserve University, USA . *Sprouts: Working Papers on Information Systems*, 5(14). http://sprouts.aisnet.org/5-14

Introduction

In our effort to understand technology, to propose its potential value, and to anticipate its use, we often find that information systems, especially those used in pervasive computing, represent an ad hoc collection of data, services and devices that have been created and recreated by users on an as-needed basis. It is difficult for designers to predict how applications will be used in pervasive computing because it is the user who ultimately defines what applications and configurations are used in which contexts. As pervasive computing becomes more prevalent, it is important that designers build systems that support these unique, in-use and user-defined characteristics.

Pervasive computing has gained increasing importance as systems continue to move toward an assemblage of distributed Internet-based services and devices that support the exchange and sharing of data and processes. Technologies such as mobile telephones, personal digital assistants and web portals are used daily in pervasive computing. These technologies allow for user expressiveness or customization around such issues as computing style, program preferences and aesthetic layout (Pask, 1971). Pervasive computing must be treated as an endless expression of varying interfaces, contexts and automation (Abowd & Mynatt, 2000; Gabriel, 2002). In this paper we explore the expressiveness within pervasive computing, proposing ways to use existing technologies to enhance this characteristic.

Many dimensions describe pervasive computing, including the convergence of technologies, the relinquishment of personal data for the receipt of particular services and the blurring of physical and virtual space (Abowd & Mynatt, 2000). We investigate and propose solutions for achieving two aspects of pervasive computing: device independence and usability (Lum & Lau, 2002). Device independence is the separation of data and presentation in support of the movement of data between technologies and heterogeneous computing systems. Device independence describes any device capable of being replaced without affecting data, services, work flow or personal computing style. Pervasive computing and its support of user-designed expressive systems require that we consider device independence, whether passing data between devices or across new versions of the same devices.

Additionally, we contend that usability affects the success of computing in general and pervasive computing in particular. Shneiderman argues that usability has become an important issue for computing research (Shneiderman, 2000, p. 86). In particular, he suggests that broad technical variety has a direct impact on usability and that speed in the distribution of data across devices must be considered. With that, we explore usability in the context of the technical variety available within pervasive computing through three measures of computing and network efficiency.

We use published interface design standards and their ability to separate requested data and its presentation to both achieve device independence and improve usability. Accessing and using the same data regardless of platform on a PDA, a laptop and a telephone demands the separation of data and presentation and we need only to revisit published interface design standards to realize these aspects of pervasive computing. Since the arrival of markup languages, we have long had the *ability* to separate the data and presentation but not the *motivation*. Pervasive computing provides the motivation. The use of published standards addresses Abowd and Mynatt (Abowd & Mynatt, 2000), who call for evaluative research of existing technologies in the design of pervasive computing and Weiser (Weiser, 1991, 1993) who notes that research on pervasive computing must ultimately be functional for both designers and users.

We begin the sections that follow by defining the role that device independence and usability play in pervasive computing. Next, we look at how device independence can be provided through interface design standards. Through an illustrative case, we show how published design standards lead to the creation of device independence and subsequently pervasive computing. We then illustrate how the same interface design standards also provide improved usability for pervasive computing. Finally, we discuss how interface design standards provide an available technology for application designers in the development and implementation of pervasive computing.

Device Independence and Usability for Pervasive Computing

Pervasive computing is a socio-technical phenomenon where devices, services and data are integrated into our daily lives (Weiser, 1991). Pervasive computing comprises numerous dimensions that exist in our everyday computing lives. Table 1 illustrates three research themes and eight dimensions that define pervasive computing (Abowd & Mynatt, 2000; Avital & Germonprez, 2003) and provide the context of our study.

Research Themes (Abowd and Mynatt, 2000)	**Computing Dimensions** (Avital and Germonprez, 2003)	**Description**
Universal Capture and Access	Mobility of Services	Services are portable for anytime-anywhere computing using wireless communication technologies
	Expanding into Virtual Spaces	Virtual presence, virtual work and virtual relationships break the limiting boundaries of physical places
	Proliferation of Task-Specialized Devices	Preconfigured devices and service agents operate in the background to carry out delegated and specialized tasks
Context Awareness	Information Transparency	Personal information is available across traditional boundaries in return for specialized services
	Device Independence	***Data are unbound from technology devices***
	Computing Automation	Autonomous systems are less dependent on human management
Natural Interfaces	Tailoring and Customization	Devices and services are modified in the context of use according to an individual's specifications or preferences
	Convergence of Technologies	Integration and unification of previously distinct digital technologies

Table 1. Research themes and computing dimensions for pervasive computing

The ability to access data across a variety of devices is a prerequisite of pervasive computing. Device independence provides users access to the same data, irrespective of the device from which the request originates. This emphasis on device independence for pervasive computing is timely as several researchers have been calling for the development of multi-device adaptive user interfaces (Grundy & Yang, 2002) and location sensitive systems (Abowd & Mynatt, 2000) in support of this requirement. We focus on device independence in the explanation of pervasive computing because of its extended use in computing today and its potential development through existing technologies. We also extend the thinking of device independence, illustrating that it can be provided not only through intelligent systems, but also through interface design standards. Using interface design standards for device independence is a more simplified approach than the "tour guide" systems developed by Cheverst et al. (Cheverst, Mitchell, & Davies, 1998) or the decision engine of Lum and Lau (Lum & Lau, 2002). These studies placed a specific emphasis on object recognition and context identification through specific devices. In this study, we investigate how to create device independence with little "real-time awareness" by any device. Instead, we propose interface design standards as a means of achieving device independence through otherwise passive and standardized design rules.

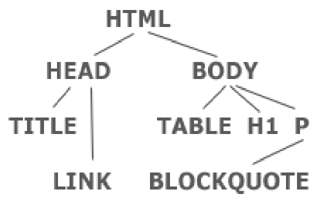
Faced with rapidly changing technologies and social demographics, it is important to understand usability for pervasive computing in the support of technical variety [5] and network access capabilities as well as the ability to accommodate a variety of users with different capabilities and disadvantages (Shneiderman, 2000). Usability in pervasive computing is translated into issues such as the delivery of data to a variety of devices with unpredictable screen sizes and technological capabilities as well as the delivery of data in a rapid manner over heterogeneous networks. We believe that interface device standards play a role in improving the delivery by allowing data to be delivered effectively to the variety of devices present in pervasive computing. In addition, while interface design standards are not converters of information, they serve to seamlessly deliver the same information to multiple devices. In this light, usability is operationalized as the speed at which data is presented (Abowd & Mynatt, 2000; Calongne, 2001). The amount of time it takes for data to be presented is a quantifiable measure that can be used to define usability and provide designers with a more concrete performance metric (Calongne, 2001; Lum & Lau, 2002; Shneiderman, 1992).

Interface Design Standards for Device Independence and Usability

Interface design standards are used for the presentation of data on the Internet. The development of interface design standards for the presentation of data began with the development of the SGML in 1978. The first working draft of the SGML standard was published in 1980; by 1985 a draft proposal for an international standard was published. Another year of review and comment resulted in the final text, which was published by CERN in 1986. SGML was too broad a language for implementation and evolved into a more limited interface design standard known as the Hypertext Markup Language (HTML). Despite its popularity, HTML has drawbacks as an interface design standard. In addition, despite the advantages that newer standards such as Cascading Style Sheets (CSS) provide, their adoption has been slow. A brief review of HTML is provided below and is followed by a review of CSS that illustrates the advances that have taken place in interface design standards.

Hyper Text Markup Language

Hypertext markup language is the language most commonly used to display data on the Internet. It is a structural markup language that describes how the different elements of a document are related.[1] Under this standard, data and presentation are located in a single document with no separation between the two. The data is formatted through tags, which control presentation characteristics such as font sizes, color and layout. A drawback of HTML is its inability to characterize data semantically. Instead, specific data about an organization a new product, or the weather is embedded in general HTML tags of <table> and <H1> (Figure 1).




```
p.title{
margin-left: +10em;
color: #1f1f1f;
font-weight: bolder;
font-size: medium;
font-family: Verdana, Helvetica;}
```

Figure 1. Illustration of HTML (left) and CSS (Right)

Cascading Style Sheets

Cascading style sheets (CSS) is a complementary interface design standard to HTML. The use of style sheets allows data within an HTML document to be separated from its presentation. Effectively combining the use of CSS with HTML is dependent on the existing structure of HTML (Figure 1). The combination allows for device independence because the presentation and device specific data are described by the CSS (Lie & Saarela, 1999) and not the HTML. From a user perspective, style sheets have a negligible impact on how data is displayed on the same device. Figure 2 is an example of how CSS handles the display of data on the same device (a desktop web client). In the example, CSS was applied to the HTML data page generated by an Ebay auction. The definition of the margins and tables were pulled out of the HTML and placed in CSS.

Ebay Page with only HTML

Ebay Page with HTML and CSS

Figure 2. Display similarity when using CSS

[1] http://www.mako4css.com/

Research Model

Interface design standards are expected to improve device independence by reducing the number of images, the amount of tabling and the redundancy of presentation rules. Specifically, interface design standards are used to explicitly provide device independence by unbinding the presentation of data from any single device. The standards are also expected to have an impact on usability as they improve both network and computer efficiencies (Figure 3).

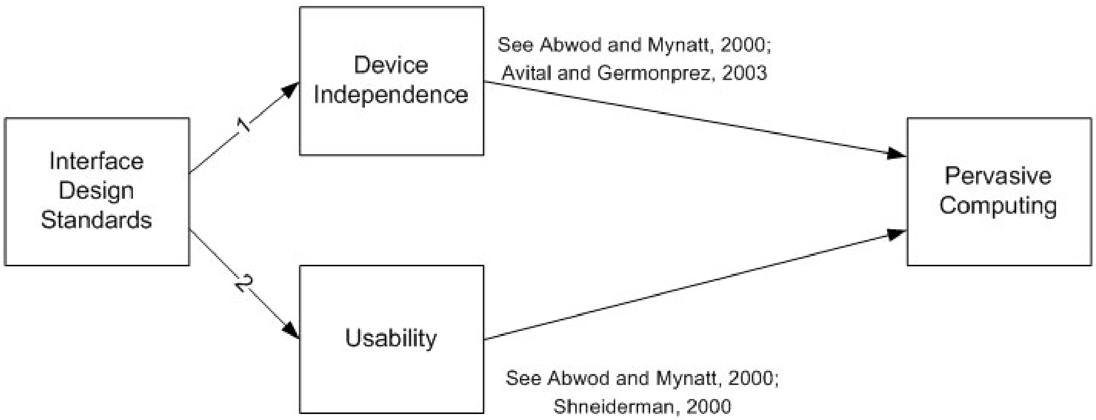


Figure 3. Research model: Interface design standards for device independence and usability

Device independence is illustrated in more detail through the redesign of one web site, which shows how interface design standards improve the presentation of data across devices, therefore supporting device independence (Arrow 1 in Figure 3). This provides our first research question:

Question 1: Can interface design standards be used to promote device independence?

In addition, through the interface design standards, page sizes can be consistently reduced by roughly 50%. For example, using the standards, the aforementioned Ebay page (Figure 3) was reduced from 20Kb to 11Kb. We expect this reduction in page size to result in usability improvements, as represented by three measures (Arrow 2 in Figure 3). This provides our second, third and fourth research questions.

Question 2: What impact do interface design standards and the subsequent reduction of page size have on the amount of time it takes for a client to display data?

Question 3: What is the impact of interface design standards and the subsequent reduction in page size on CPU resources consumed by clients?

Question 4: What is the impact of interface design standards and the subsequent reduction in the page size on CPU resources consumed by servers?

Research Findings

Two explanations are provided in the examination of the impact that interface design standards have on the display and delivery of data. First, we use a descriptive explanation in an illustrative case to show how interface design standards are used to provide device independence (Question 1). Second, we use a statistical relevance explanation in an experiment to illustrate how interface design standards improve usability through three efficiency variables - the rendering time of the client application, the client CPU utilization and the server CPU utilization (Questions 2-4).

Impact of Interface Design Standards for Device Independence

One of the most powerful characteristics of interface design standards is their management of data presentation. HTML tables have generally been the most common presentation vehicles. Specifically, data is presented in a series of tables, embedded tables and cells. In Figure 4, Association of Computing Machinery (ACM) data is presented using HTML tables. In the figure, the first image is the ACM web site at the desktop; the second image is the same site with the tables highlighted. The colors of the table borders are reflective of nesting order: outer tables are blue, second-level tables are green and third-level and deeper tables are red.[2]

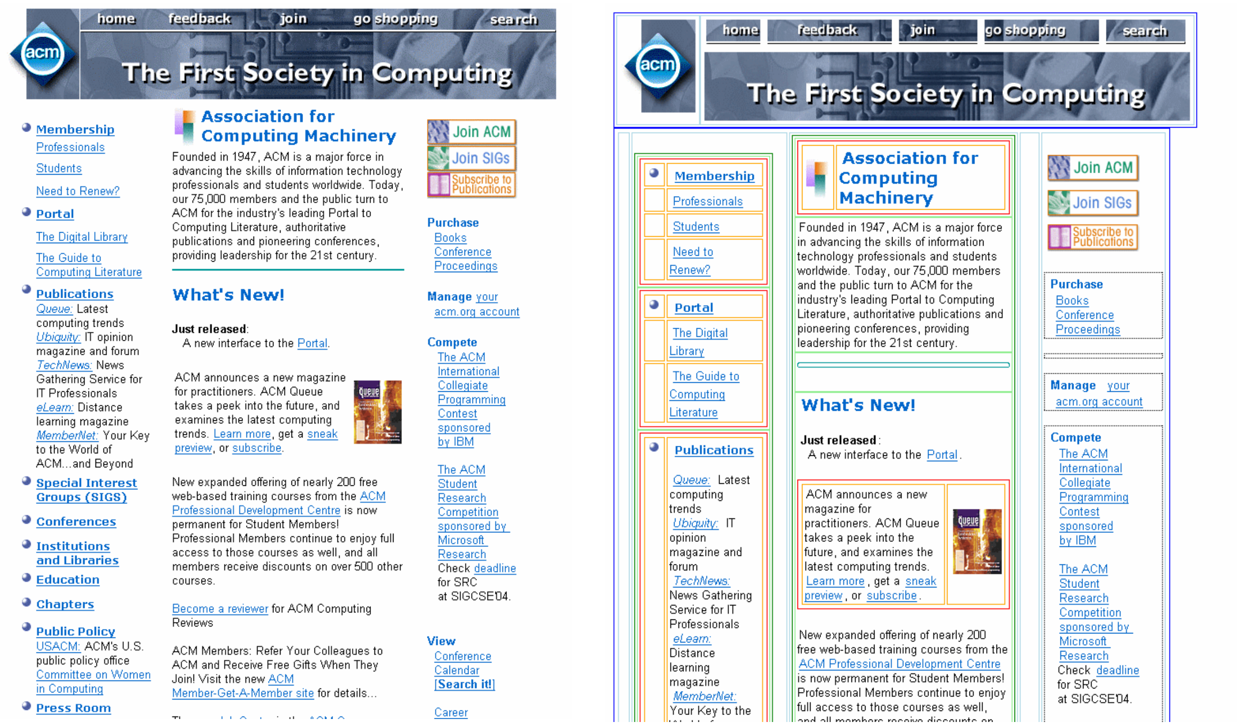


Figure 4. Site through desktop using HTML tables

As shown, the site relies heavily on HTML tables, a technique that works well only when displaying data on a desktop. When table structures are accessed through a smaller device such

[2] http://www.squarefree.com/bookmarklets/webdevel.html#show_blocks

as a PDA, the presentation of data cannot adapt to the smaller interface space. Figure 5 shows a segment of the same site accessed through a PDA.

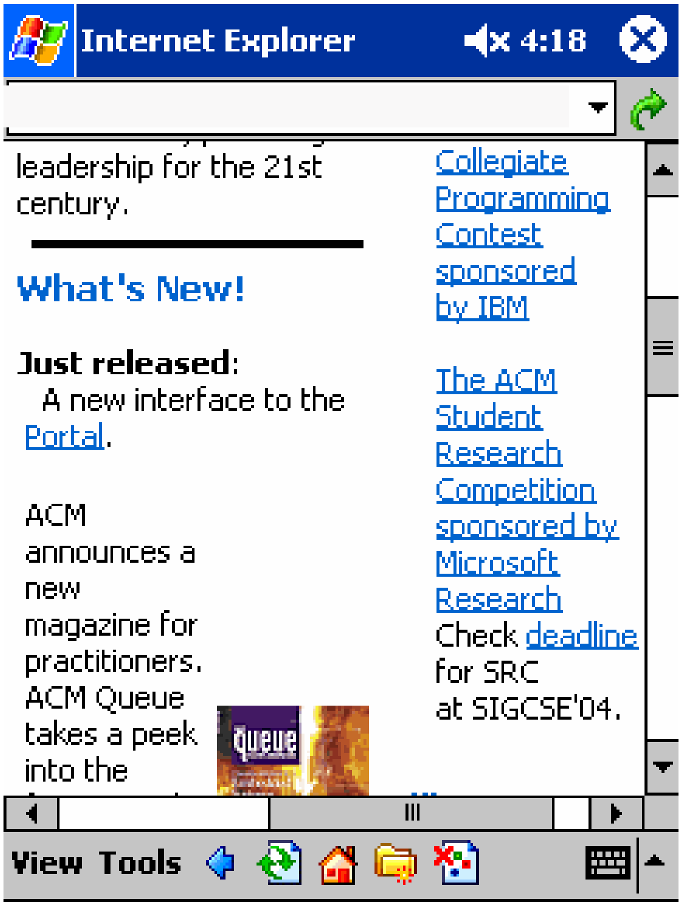


Figure 5. Site through PDA using HTML tables

As seen in Figure 5, the exact presentation is maintained in the PDA, even though it provides a smaller window through which the same data is viewed. Both vertical and horizontal scrolling are required as the presentation of data maintains its large screen design. In essence, presentation through HTML tables treats the PDA as a small lens that is moved up and down and left and right across the larger canvas of a desktop interface. In this case, the system did not scale across devices and thus failed to provide support for a broad suite of devices needed for pervasive computing (Abowd & Mynatt, 2000). Device independence failed through a breakdown of the natural interfaces the PDA provided.

The same site can be restructured to remove many of the HTML tables. Using the interface design standard of CSS, presentation occurs through HTML markers, not tables. That is, data is categorized based on its relation to other data, not its absolute position on a page. Figure 6 shows the presentation of the site using CSS. The first image is the earlier HTML designed site and the second is the site redesigned using HTML and CSS.

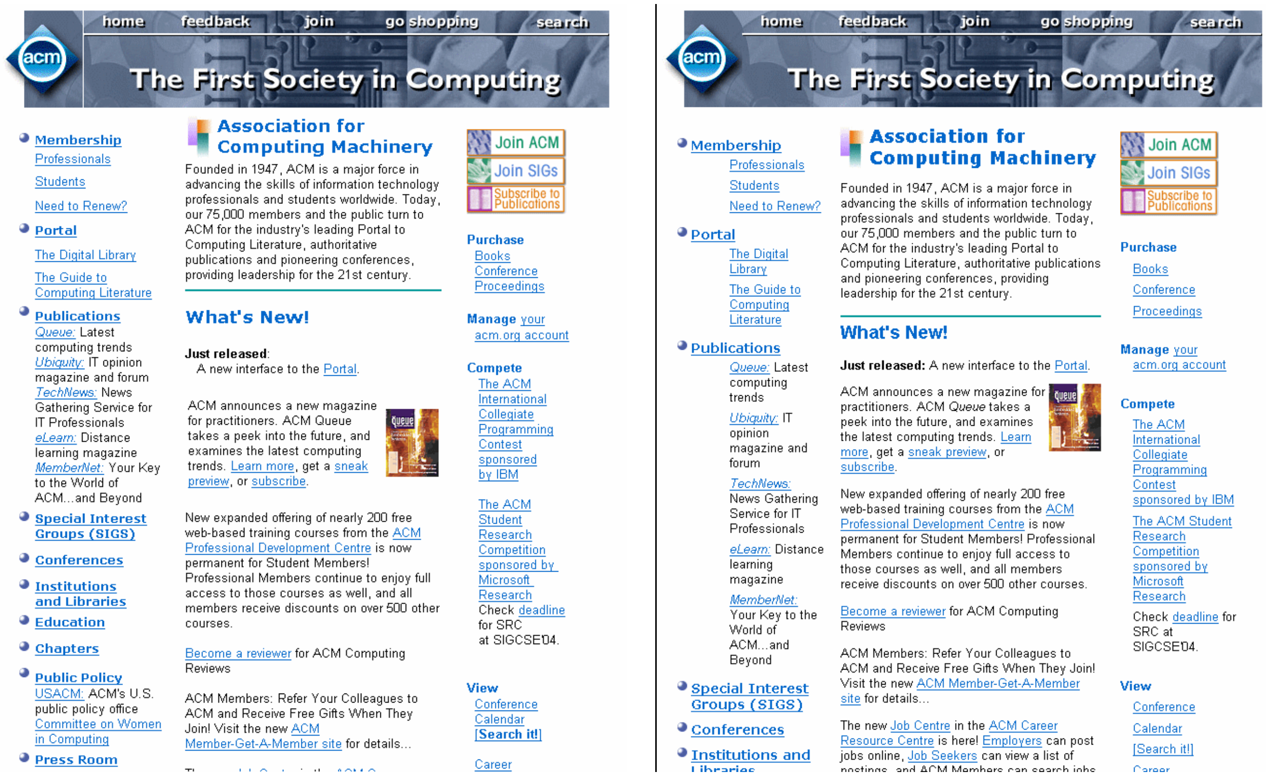


Figure 6. Site similarity using interface design standards

The differences that interface design standards provide via the desktop are visually insignificant. However, a closer look at the site through a PDA shows that the data is rendered on the fly in order to accommodate the new device. Figure 7 shows the site displayed through a PDA after being redesigned using HTML and the interface design standard of CSS.

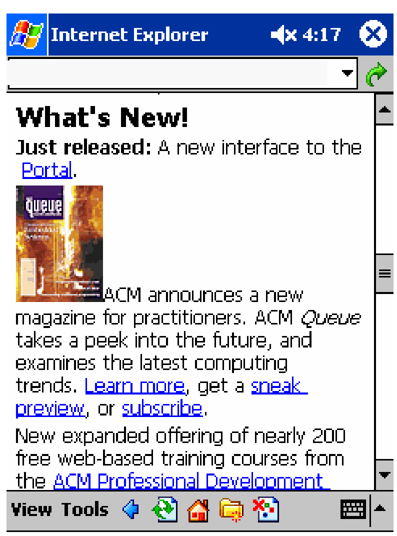


Figure 7. Web site through PDA using interface design standards

Using HTML *and* CSS, the presentation becomes entirely vertical without any horizontal navigation. Interface design standards present the data sequentially by removing the majority of HTML tables that previously fixed the width of the site. In this context, the site is presented equally well on a large space like a desktop and the small space of a PDA. In fact, the presentation of the data is more accommodating to the natural interface of the PDA. In separating the data and presentation, we were able to achieve an improvement in pervasive computing device independence without altering the intelligence of any device. Instead, we promote device independence by adhering to the natural interfaces of multiple devices through the otherwise passive technology of CSS (Question 1).

Impact of Interface Design Standards for Usability

As mentioned, an improvement provided by interface design standards was a reduction in page size. The impact of reduced page size is examined on three usability outcomes. First, we investigate the role that reduced page size plays in determining the amount of time required for a client to render an interface. Rendering time is an important issue in usability research and pervasive computing as users move across devices with varying levels of processing power. Second, servers and clients use computing resources to deliver and display data. The effect of page size is explored with respect to the reduction of resources that servers use in distributing data and the decrease in resources clients use to display it. Resource utilization is an important usability issue in pervasive computing as users place an increasing burden on clients and servers to deliver an increasingly expansive set of data and services.

In the design of the experiment to investigate these questions, the data server and the clients were run on the same device, disconnected from the main network to avoid the interference of network parameters. Caching was also disabled among clients and within the server since complete page loads from the server, not local copies, were required. Finally, all interface design code (the CSS) was embedded into the HTML and not in a separate file. This was accomplished to illustrate that interface design standards can reduce page size without simply splitting a file into two parts. The experiment made use of two popular desktop web browsers: Netscape Navigator and Internet Explorer. The use of two clients was to see if the different rendering engines of the clients showed similar improvements in rendering times. Additionally the clients were treated independently and not as a combined representation of a web client since rendering times and client CPU utilization were shown to be significantly different between the two.

The data used and served from the server to the clients were 3 webpages that were used in their original HTML format and then reformatted using HTML and CSS (Questions 2-4). Additionally, the redesigned web pages are aesthetically the same as the originals (see Figures 2 and 6). As expected, size was reduced using interface design standards across all three pages (Table 2).

Association of Computing Machinery		**E-Bay**		**Los Alamos National Laboratory**	
Size of HTML Only	Size of HTML and CSS	Size of HTML Only	Size of HTML and CSS	Size of HTML Only	Size of HTML and CSS
26 KB	13 KB	20 KB	11 KB	21 KB	11 KB

Table 2. Tested web sites and reduced page size

With regard to the three efficiency variables, rendering time was measured from the start of the page load to the completion of the last character on the page, providing an accurate capture of the load time for the page. The measurement of CPU utilization for both client and server was done using a performance monitor application. Fifty samples were taken to gather rendering times and resource utilization on the client and server (N=50) for every web site.

Table 3 shows the results of the t-tests that suggest improved efficiencies from a HTML page to an HTML/CSS formatted page. The table represents percentage improvements in CPU load times (for both the client and the server) and the rendering times when using the HTML/CSS reformatted pages. Significance indicates that HTML/CSS reformatted pages statistically improved the associated measures. With respect to Question 2, regarding the impact of CSS and the subsequent reduction of page size on the amount of time it takes for a client to display data, rendering time was significantly reduced in all cases for both the desktop and PDA. With respect to Questions 3 and 4, regarding the impact on CPU resources, the effects of the CSS in reducing the CPU utilization was mixed in both the client and the server.

	Mozilla Netscape			*Internet Explorer*			
	CPU Load: Client	CPU Load: Server	Rendering Time (Desktop)	CPU Load: Client	CPU Load: Server	Rendering Time (Desktop)	Rendering Time (PDA)
ACM	(9.0%)	8.0%	33.0%*	4.7%	(1.5%)	30.0%*	41.3%*
eBay	13.9%*	18.9%*	36.8%*	30.9%*	21.7%*	63.4%*	N/A
Los Alamos	12.2%*	35.7%*	18.2%*	10.8%*	16.2%*	47.6%*	79.6%*

* Indicates Significance
( ) Indicates a decrease in performance
Values represent percentage improvements in CPU load times and the rendering times when using the HTML/CSS reformatted pages.

Table 3. Results of CSS on resource utilization and rendering time

In the experiment, each page (in both its original and redesigned state) was loaded into two popular web browsers. As seen, there was no significant improvement in the utilization of client or server resources due to the reduction of page size for the ACM web page. Regarding this, we are unsure as to why this was the case. We noticed that the initial rendering times of the ACM page were faster than those of Los Alamos or eBay and believe that the reduced rendering time using CSS were simply not enough to provide significant changes in CPU load time. The other two sites, however, showed an improvement in the CPU utilization for both the client and server processes.

Discussion

Interface design standards have been shown to play an important role in pervasive computing. This paper illustrates that there are multiple benefits to using interface design standards, two of which include device independence and improved usability. In pervasive computing, these issues play a significant role in determining how data is *delivered* and *displayed* across various devices. As users are capable of surveying and selecting data that suits their current needs, system designers must be aware that the delivery and display of data must be as streamlined and dynamic as possible. Serving data to an unknown suite of devices including desktops, handhelds and wireless telephones increases the need for device independence and efficiency in the delivery and rendering of data. Interface design standards have been shown to alleviate both of these concerns by improving usability and device independence.

Interface design standards provide one of several methods for improving device independence and usability. In the case of device independence, different HTML pages can be served based on server recognition of a client operating system and browser. Numerous web sites rely on different pages served to different devices based on browsers, operating systems and connection speeds (Lum & Lau, 2002). With respect to usability, technologies such as caching and data distribution networks provide improved data delivery and rendering speeds. In any case, interface design standards are capable of complementing these alternate technologies in the effective delivery and presentation of data.

The impact of interface design standards on pervasive computing should not be underestimated. As pervasive computing devices proliferate and become used on a mass scale, networks are going to become burdened with data traveling between providers and the users. Providers have the opportunity to improve the efficiency of their servers and network connections by making the data that they serve richer in experience and smaller in size. Additionally, providers must maintain device independence as their data and services are consumed across various devices. The use of CSS as a standard for developing data on the web for pervasive computing devices will allow providers to take advantage of the multi-faceted nature of interface design standards.

Acknowledgements

The authors would like to acknowledge Eric Meyer and Pankaj Nagpal for their help in this paper.

References

Abowd, G., & Mynatt, E. (2000). *Charting Past, Present, and Future Research in Ubiquitous Computing* (Vol. 7).

Avital, M., & Germonprez, M. (2003). *Ubiquitous Computing: Surfing the Trend in a Balanced Act.* Cleveland Ohio.

Calongne, C. M. (2001). *Designing for web site usability.* Paper presented at the The Journal of Computing in Small Colleges.

Cheverst, K., Mitchell, K., & Davies, N. (1998). Design of an Object Model for a Context Sensitive Tour Guide, *Conference on Interactive Applications of Mobile Computing*. Rostock, Germany.

Gabriel, Y. (2002). Essay: On Paragrammatic Uses of Organizational Theory - A Provocation. *Organization Studies, 23*, 133-151.

Grundy, J., & Yang, B. (2002). An environment for developing adaptive, multi-device user interfaces. In *Fourth Australasian User Interface Conference*. Adelaide.

Lie, H. W., & Saarela, J. (1999, Oct). Multipurpose Web publishing using HTML, XML, and CSS. *Communications of the ACM, 42,* 95-101.

Lum, W., & Lau, F. (2002). A Context-Aware Decision Engine for Content Adaptation, *IEEE Pervasive Computing* (Vol. 1, pp. 41-49).

Pask, G. (Ed.). (1971). *A Comment, a Case History, and a Plan*. London: Studio Vista.

Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading MA: Addison-Wesley.

Shneiderman, B. (Cartographer). (2000). *Universal Usability*

Weiser, M. (1991). The computer for the 21st century. On *Scientific American*.

Weiser, M. (Artist). (1993). *Some computer science issues in ubiquitous computing*

芽|Sprouts

Working Papers on Information Systems | ISSN 1535-6078

Office:
Sprouts
University of Amsterdam
Roetersstraat 11, Room E 2.74
1018 WB Amsterdam, Netherlands
Email: admin@sprouts.aisnet.org